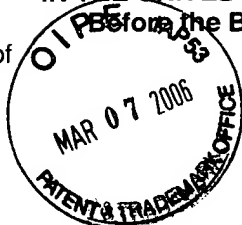


IN THE UNITED STATES PATENT AND TRADEMARK OFFICE



In re Patent Application of

✓ O'BRIEN et al

Serial No. 09/043,406

Filed: March 18, 1998

Title: SERVICE PROVISION SYSTEM FOR  
USE IN DISTRIBUTED PROCESSING ENVIRONMENTS

**Mail Stop Appeal Brief - Patents**

Commissioner for Patents

P.O. Box 1450

Alexandria, VA 22313-1450

Atty Dkt. 36-1148

C# M#

TC/A.U.: 3639

Examiner: Robinson, A.

Date: March 7, 2006

AP#  
Ifw

Sir:

☐ Correspondence Address Indication Form Attached.

☐ **NOTICE OF APPEAL**

Applicant hereby **appeals** to the Board of Patent Appeals and Interferences  
from the last decision of the Examiner twice/finally rejecting  
applicant's claim(s).

\$500.00 (1401)/\$250.00 (2401) \$

☒ An appeal **BRIEF** is attached in the pending appeal of the  
above-identified application

\$500.00 (1402)/\$250.00 (2402) \$ 500.00

☐ Credit for fees paid in prior appeal without decision on merits

-\$ ( )

☐ A reply brief is attached.

(no fee)

☒ Petition is hereby made to extend the current due date so as to cover the filing date of this  
paper and attachment(s)

One Month Extension \$120.00 (1251)/\$60.00 (2251)  
Two Month Extensions \$450.00 (1252)/\$225.00 (2252)  
Three Month Extensions \$1020.00 (1253)/\$510.00 (2253)  
Four Month Extensions \$1590.00 (1254)/\$795.00 (2254)

\$ 450.00

☐ "Small entity" statement attached.

Less month extension previously paid on

-\$ ( )

**TOTAL FEE ENCLOSED \$ 950.00**

Any future submission requiring an extension of time is hereby stated to include a petition for such time extension.  
The Commissioner is hereby authorized to charge any deficiency, or credit any overpayment, in the fee(s) filed, or  
asserted to be filed, or which should have been filed herewith (or with any paper hereafter filed in this application by this  
firm) to our **Account No. 14-1140**. A duplicate copy of this sheet is attached.

901 North Glebe Road, 11th Floor  
Arlington, Virginia 22203-1808  
Telephone: (703) 816-4000  
Facsimile: (703) 816-4100  
RYM:sl

NIXON & VANDERHYE P.C.

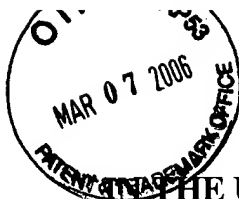
By Atty: Raymond Y. Mah, Reg. No. 11,426

03/08/2006 5:23 PM E1 00000018 09043406

01 FC:1252

450.00 OP

Signature:



THE UNITED STATES PATENT AND TRADEMARK OFFICE  
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Patent Application of

O'BRIEN et al

Atty. Ref.: 36-1148

Serial No. 09/043,406

TC/A.U.: 3639

Filed: March 18, 1998

Examiner: Robinson, A.

For: SERVICE PROVISION SYSTEM FOR

USE IN DISTRIBUTED PROCESSING ENVIRONMENTS

\*\*\*\*\*

March 7, 2006

Mail Stop Appeal Brief - Patents  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

**APPEAL BRIEF**

Sir:

Appellant hereby **appeals** to the Board of Patent Appeals and Interferences from  
the last decision of the Examiner.

03/08/2006 SZEWDIE1 00000018 09043406

02 FC:1402

500.00 0P

**TABLE OF CONTENTS**

(I)	REAL PARTY IN INTEREST .....	3
(II)	RELATED APPEALS AND INTERFERENCES.....	4
(III)	STATUS OF CLAIMS .....	5
(IV)	STATUS OF AMENDMENTS .....	6
(V)	SUMMARY OF CLAIMED SUBJECT MATTER .....	7
(VI)	GROUND OF REJECTION TO BE REVIEWED ON APPEAL.....	16
(VII)	ARGUMENT .....	17
(VIII)	CLAIMS APPENDIX .....	27
(IX)	EVIDENCE APPENDIX .....	35
(X)	RELATED PROCEEDINGS APPENDIX .....	36

(I) **REAL PARTY IN INTEREST**

The real party in interest is British Telecommunications public limited company, a corporation of the country of England.

(II) **RELATED APPEALS AND INTERFERENCES**

The appellant, the undersigned, and the assignee are not aware of any related appeals, interferences, or judicial proceedings (past or present), which will directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

**(III) STATUS OF CLAIMS**

Claims 53-58 and 61-76 are pending and have been rejected. No claims have been substantively allowed. While the first sentence of section 5 of the Final Rejection states that “Claims 53-58 and 61-66 are rejected under 35 U.S.C. 103(a) as being unpatentable over Parad (US 5,369,570), in further view of Wrabetz et al (US 5,442,791) (emphasis added),” Appellant assumes that the Examiner intended to include claim 67 in this rejection.

**(IV) STATUS OF AMENDMENTS**

No amendments have been filed since the date of the Final Rejection. However, Appellant now makes note of an error in the dependency of (apparatus) claim 75 which presently indicates a dependency from (method) claim 72, but which should correctly depend instead from (apparatus) claim 74. The rejection of dependent claim 75 stands or falls with the rejection of its base claim, and thus for purposes of this appeal, Appellant respectfully requests that an amendment to claim 75 (i.e., changing its dependency from claim 72 to 74) be held in abeyance until disposition of the appeal.

(V) **SUMMARY OF CLAIMED SUBJECT MATTER**

The invention of the claims relates to a service provision system/method for use in distributed processing environments. A listing of each independent claim, each dependent claim argued separately and each claim having means plus function language is provided below including exemplary reference(s) to page and line number(s) of the specification.

53. A service provision system [pg. 10, l. 20 - pg. 11, l. 7] for use in distributed processing environments [pg. 10, l. 20-27], said system comprising:

an input for receiving a service request [pg. 10, l. 32 to pg. 11, l. 1; pg. 20, ll. 1-4];

processing means for processing the service request [pg. 12, ll. 23-26];

negotiation means for use in establishing conditions applicable to provision of one or more component processes involved in provision of the service, said negotiation means being adapted to assemble said conditions proactively by negotiation prior to receipt of said service request [pg. 15, ll. 3-7 and 25-27; pg. 16, l. 22 to pg. 17, l. 2];

an up-datable data store [pg. 12, l. 19-22];

means to access said up-datable data store for storing said conditions when established and assembled [pg. 11, ll. 1-5; pg. 12, l. 19-22]; and

an output for providing a response to the service request, said response comprising an indication of availability of the requested service [pg. 11, ll. 3-5; pg. 20, ll. 1-4];

wherein the processing means is adapted to process a service request by accessing one or more of the previously established conditions in the data store, processing the request using the one or more established conditions and producing said response [pg. 10, l. 32 to pg. 11, l. 5; pg. 12, ll. 23-26; pg. 13, ll. 1-5].



54. A service provision system as in claim 53 wherein one or more of said established conditions has an associated expiry time after which it is no longer applicable [pg. 17, ll. 3-10; pg. 19, ll. 7-13].

55. A service provision system as in claim 54 wherein the processing means is adapted to detect an expired or undefined condition in the data store, which condition is applicable to a component process for the provision of a requested service, and to trigger the negotiation means to establish a substitute condition [pg. 4, ll. 1-5; pg. 7, ll. 15-24].

56. A service provision system as in claim 55 further comprising:  
means to access said data store for storing data related to services offered by the system and to one or more entities which have an interest in receiving information relating to one or more of said services, together with means to transmit information based on said data related to services to the one or more entities which have an interest [pg. 5, ll. 3-17; pg. 6, ll. 12-14; pg. 7, ll. 15-24; pg. 11, ll. 1-5; pg. 20, ll. 1-4].

57. A system as in claim 53 which further comprises initiation means to initiate one or more component processes in provision of a requested service [pg. 20, ll. 5-10].

58. A service provision system as in claim 57 wherein:  
provisioning a requested service requires provision of a selected set of component processes;

the negotiation means establishes and stores a set of conditions applicable to provision of the component processes of the selected set; and

the processing means is adapted to process a service request by accessing the stored set of conditions in the data store, processing the request using said stored set, and producing said response [pg. 4, l. 28 to pg. 5, l. 1; pg. 7, ll. 7-14].

61. A service provision method [pg. 10, l. 20 - pg. 11, l. 7] for use in distributed processing environments [pg. 10, l. 20-27], said method comprising:

establishing conditions applicable to provision of one or more component processes in a service, proactively by negotiation prior to receipt of said service request [pg. 15, ll. 3-7 and 25-27; pg. 16, l. 22 to pg. 17, l. 2];

accessing an up-datable data store and storing said conditions once established [pg. 11, ll. 1-5; pg. 12, l. 19-22];

subsequently receiving a request for said service [pg. 10, l. 32 to pg. 11, l. 1; pg. 20, ll. 1-4];

processing said service request by:

a) accessing one or more of said previously established conditions in the data store [pg. 11, ll. 1-5; pg. 12, l. 19-22]; and

b) providing a response to the service request, said response comprising an indication of availability of the requested service dependent upon whether said one or more established conditions is met [pg.10, l. 32 to pg. 11, l. 5; pg. 12, ll. 23-26; pg. 13, ll. 1-5].

62. A service provision method according to claim 61 wherein one or more of said established conditions stored in said data store is applicable until advent of an expiry time associated with said one or more conditions [pg. 17, ll. 3-10; pg. 19, ll. 7-13].

63. A service provision method according to claim 62 wherein further comprising the step, responsive to receipt of said request, of finding whether any conditions for provision of said service are extant and substituting a substitute condition in the event that no such conditions are found [pg. 4, ll. 1-5; pg. 7, ll. 15-24].

64. A service provision method according to claim 61 wherein said method further comprises the step of scheduling provision of said one or more component processes, said step being carried out after receipt of said request for said service [pg. 5, ll. 3-17; pg. 6, ll. 12-14; pg. 7, ll. 15-24; pg. 11, ll. 1-5].

65. A service provision method according to claim 64 wherein said method further comprises the step, responsive to a failure to schedule one or more component processes, of carrying out one of the following steps:

- i) re-schedule the component process;
- ii) transmit a message to an entity which requested the service indicating that the service can only be provided under conditions different to said previously established conditions;
- iii) re-assign the service to another service provider; or

iv) indicate to an entity which requested the service that the requested service cannot be provided [pg. 5, ll. 3-17; pg. 6, ll. 12-14; pg. 7, ll. 15-24; pg. 11, ll. 1-5].

66. A method according to claim 61 further comprising the step of identifying component processes for use in provisioning the requested service [pg. 2, l. 32 to pg. 3, l. 2; pg. 14, ll. 25-33].

67. A method according to claim 66 which further comprises initiating one or more of said component processes identified for use in the requested service [pg. 20, ll. 5-10].

68. A method of operating one or more computers to manage a business process, said method comprising:

executing a first autonomous software process [pg. 10, l. 32 to pg. 11, l. 13; pg. 12 to pg. 13, l. 19];

executing a second autonomous software process [pg. 10, l. 32 to pg. 11, l. 13; pg. 12 to pg. 13, l. 19];

the execution of said autonomous software processes establishing conditions applicable to provision of said service by negotiation between said first and second autonomous software processes [pg. 11, ll. 14-25; pg. 13, ll. 1-12; pg. 16, l. 11 to pg. 17, l. 2];

accessing an up-datable data store and storing said conditions in said data store once established [pg. 12, ll. 19-22];

the execution of said first autonomous software process subsequently involving the receipt of a request for said service for handling by said first autonomous software process [pg. 10, l. 32 to pg. 11, l. 1; pg. 20, ll. 1-4; pg. 15, ll. 3-7]; and

responsive to receipt of said request, said first autonomous software process testing whether the established conditions for provision of said service are met and providing an indication as to whether the requested service is available in dependence upon the outcome of said test [pg. 17, ll. 3-28].

69. A method according to claim 68 in which one or more of said established conditions is stored in said data store until advent of an expiry time associated with said one or more conditions [pg. 17, ll. 3-10; pg. 19, ll. 7-13].

70. A method according to claim 68 wherein, responsive to receipt of said request, said first autonomous software process tests whether any conditions for provision of said service are extant and substitutes a substitute condition in the event that no such conditions are found [pg. 4, ll. 1-5; pg. 7, ll. 15-24].

71. A method according to claim 69 wherein, responsive to receipt of said request, said first autonomous software process tests whether any conditions for provision of said service are extant and substitutes a substitute condition in the event that no such conditions are found [pg. 4, ll. 1-5; pg. 7, ll. 15-24].

72. A method according to claim 68 in which said first autonomous software process identifies one or more further autonomous software processes representing resources required to provide one or more component services in the provision of said service, said method further comprising executing said further autonomous software process representing a component service provider [pg. 2, l. 32 to pg. 3, l. 2; pg. 14, ll. 25-33].

73. A method according to claim 69 in which said first autonomous software process identifies one or more further autonomous software processes representing resources required to provide one or more component services in the provision of said service, said method further comprising executing said further autonomous software process representing a component service provider [pg. 2, l. 32 to pg. 3, l. 2; pg. 14, ll. 25-33].

74. An apparatus for representing a provider of a service component of a business process, said apparatus comprising [pg. 10, l. 20 to pg. 11, l. 5; pg. 13, ll. 1-14]:

a computer having a memory storing autonomous software code executable to provide an autonomous software process representing a service requester, the execution of said autonomous software process [Figs. 1 and 8; pg. 10, l. 32 to pg. 11, l. 13; pg. 12 to pg. 13, l. 19]:

a) establishing conditions applicable to provision of said service by negotiation between said autonomous software process and one or more other

autonomous software processes [pg. 11, ll. 14-25; pg. 13, ll. 1-12; pg. 15, ll. 37 and 25-27; pg. 16, l. 11 to pg. 17, l. 2];

b) accessing an up-datable data store and storing said conditions in said data store once established [pg. 11, ll. 1-5; pg. 12, ll. 19-22]; and

c) on subsequently receiving a request for said service, accessing said data store and testing whether the established conditions for provision of said service are met and providing an indication as to whether the requested service is available in dependence upon the outcome of said test [pg. 15, ll. 3-7; pg. 17, ll. 3-28].

75. Apparatus according to claim 72 in which said autonomous software code is executable to provide a plurality of software threads, able to concurrently interact with different other autonomous software processes [pg. 8, ll. 5-9; pg. 13, ll. 2-4; pg. 22, ll. 26-31].

76. A method of operating one or more computers to manage a business process, said method comprising:

executing a first autonomous software process [pg. 10, l. 32 to pg. 11, l. 13; pg. 12 to pg. 13, l. 19];

executing a second autonomous software process [pg. 10, l. 32 to pg. 11, l. 13; pg. 12 to pg. 13, l. 19];

the execution of said autonomous software processes establishing conditions applicable to provision of said service by negotiation between said first and second

autonomous software processes [pg. 11, ll. 14-25; pg. 13, ll. 1-12; pg. 16, l. 11 to pg. 17, l. 2];

accessing an up-datable data store and storing said conditions in said data store once established [pg. 12, ll. 19-22];

the execution of said first autonomous software process subsequently involving the receipt of a request for said service for handling by said first autonomous software process [pg. 10, l. 32 to pg. 11, l. 1; pg. 20, ll. 1-4; pg. 15, ll. 3-7]; and

responsive to receipt of said request, said first autonomous software process testing whether the established conditions for provision of said service are met and providing an indication as to whether the requested service is available in dependence upon the outcome of said test [pg. 17, ll. 3-28];

wherein there are no control dependencies between the first and second autonomous software processes [pg. 11, ll. 16-19].



**(VI) GROUND OF REJECTION TO BE REVIEWED ON APPEAL**

Claims 68 and 76 stand rejected under 35 U.S.C. §112 as allegedly being indefinite for failing to particularly point out and distinctly claim the subject matter which Appellant regards as the invention.

Claims 53-58 and 61-66 stand rejected under 35 U.S.C. §103 as allegedly being unpatentable over Parad et al. (U.S. Patent No. 5,369,570, hereinafter "Parad") in view of Wrabetz et al. (U.S. Patent No. 5,442, 791, hereinafter "Wrabetz").

Claims 68-76 stand rejected under 35 U.S.C. §103 as allegedly being unpatentable over Parad in view of Babayev et al. (U.S. Patent No. 5,615,121, hereinafter "Babayev").

**(VII) ARGUMENT**

Claims 68 and 76 are not indefinite under 35 U.S.C. §112 for failing to particularly point out and distinctly claim the subject matter which Appellant regards as the invention.

Section 3 (pages 2-3) of the final Office Action states the following:

“The terms ‘first autonomous software process’ and ‘second autonomous software process’ in claims 68 and 76 are relative terms that render the claims indefinite. The terms ‘first autonomous software process’ and ‘second autonomous software process’ are not defined by the claim, the specification does not provide a standard for ascertaining the requisite degree, and one of ordinary skill in the art would not be reasonably apprised of the scope of the invention. Because the terms ‘first autonomous software process’ and ‘second autonomous software process’ are used, both claims 68 and 76, are unclear, thereby making the scope of the invention unclear as well.”

The recitation of the terms “first autonomous software process” and “second autonomous software process” in claims 68 and 76 are not indefinite. As would be clearly understood by one of ordinary skill in the art, the “second autonomous software process” indicates another autonomous software process in addition to the “first autonomous software process.” The plain language of claims 68 and 76 thus establishes a clear relative relationship. Moreover, claims 68 and 76 further recite “...establishing conditions applicable to provision of said service by negotiation between said first and second autonomous software processes.” Accordingly, not only is a clear relative relationship established (a second autonomous software process indicating another autonomous software process with respect to the first autonomous software process), the plain language of claims 68 and 76 also clearly establishes interaction between the first and second autonomous software processes.

Moreover, the Examiner's allegation that "The specification does not provide a standard for ascertaining the requisite degree" is clearly unfounded. For example, page 10, line 32 to page 15, line 8 of the specification supports the terms first and second autonomous software processes. In particular, portions of the above-noted passage of the specification state "In this description the term 'agent' relates to a function or process which operates in the distributed computing environment and which can act autonomously to receive a request for an operation and provide a result" (page 10, line 32 to page 11, line 1), "[T]ypically an agent 10 is embodied as a piece of software for example written in the C programming language, running on a suitable computing platform 80" (page 12, lines 12-14), and "The agents 10 are able to communicate and negotiate with each other for provision of services" (page 13, lines 1-2). Consistent with the plain language of the claims, the specification thus supports the first autonomous software process and the second autonomous software process and describes a relationship therebetween.

Accordingly, Appellant respectfully requests that the rejection of claims 68 and 76 under 35 U.S.C. §112, second paragraph, be reversed.

Claims 53-58 and 61-66 are not made "obvious" under 35 U.S.C. §103 over Parad in view of Wrabetz.

In order to establish a *prima facie* case of obviousness, all of the claim limitations must be taught or suggested by the prior art. The combination of Parad and Wrabetz fails to teach or suggest all of the claim limitations. For example, the combination fails to teach or suggest "negotiation means for use in establishing conditions applicable to provision of one or more component processes involved in provision of the service, said

negotiation means being adapted to assemble said conditions proactively by negotiation prior to receipt of said service request (emphasis added),” as required by independent claim 53. Similarly, the combination also fails to teach or suggest “establishing conditions applicable to provision of one or more component processes in a service, proactively by negotiation prior to receipt of said service request (emphasis added),” as required by independent claim 61.

Section 7 (pages 16-17) of the final Office Action alleges that col. 29, lines 49-61 of Parad discloses the above claimed features. Section 5 (pages 3-4) of the final Office Action echoes this allegation. Appellant disagrees with these allegations.

In particular, section 7 of the final Office Action states, *inter alia*,

“However, in Col. 29, lines 49-54 of Parad, the Action Control negotiates with the Resource Engine. Here, the requirements are fed into the Resource Engines component where a schedule that meets requirements is attempted to be created. This attempt represents the negotiation process since the Resource Engine tries to meet specific requirements requested by the requester by matching different resources with each request.”

The requirements which “are fed into the Resource Engines component where a schedule that meets requirements is attempted to be created” (see the passage of the final Office Action noted above) do not refer to any interaction between the action control and the resource engine, let alone any alleged negotiation between the action control and the resource engine. Instead, the requirements fed into the resource engine’s component where a schedule that meets requirements may be created comes from the external enterprise information system 214 as illustrated by arrow 215. These “requirements” are not fed into the resource engine from the action control.

In more detail, arrow 215 in Fig. 2 of Parad is labeled as “REQUIREMENTS PRODUCT/PROCESS DEPENDENCIES STATUS.” Also, col. 9, lines 45-47 of Parad makes reference to “Messages describe resource requirements, products and process characteristics, dependencies and status.” It is these resource requirements from external enterprise information systems 214 to resource engine 203 which form the requirements which are fed into the resource engine 203 where a schedule meeting those fed requirements may be created. Again, this interaction between the external enterprise information systems 214 and the resource engine 203 does not involve interaction between the action control 202 and the resource engine 203, let alone any negotiation between the action control 202 and the resource engine 203. The two sentences noted above in section 7 of the final Office Action (“However, in Col. 29, lines 49-54 of Parad, the action control negotiates with the resource engine. Here, the requirements are fed into the resource engines component where a schedule that meets requirements is attempted to be created.”) therefore form a complete *non-sequitur* as the “requirements” noted in the second sentence have nothing to do with any alleged negotiation between the action control and the resource engine noted in the first sentence.

Each action control 202 disclosed by Parad is a user-interface component, whereas the resource engine 203 is a component that performs calculations of how to meet requirements presented to it from the enterprise information systems 214. The resource engine 203 does not engage in any negotiation. In particular, rules 204 from the action control 202 to the resource engine 203 are rules that are not open to negotiation. They are commands by the human manager of the resource.

Col. 29, lines 49-54 of Parad states the following:

“The strategy used by the resource engine 601 to dispatch notices is to send a single message summarizing a schedule’s condition when it changes significantly. The same strategy is employed by the action control 901.”

The first of the above two sentences merely discloses that resource engines send notices to action controls when that resource engine’s schedule changes significantly. Sending notices only in reaction to a significant change in the schedule associated with the resource presumably is advantageous because it reduces the amount of messages being sent around the system. A consequential reduction in processing performed in reaction to the notices also results.

The second of the two sentences stated above indicates that the action control only tells a (human) user that the action list is updated when that update is significant. Indeed, the sentence following col. 29, lines 49-54 of Parad states “The user (operator) is only notified when the contents of the action list changes significantly.” This portion of Parad thus discloses that the action control sends a message to the user, not to the resource engine. This portion of Parad therefore does not disclose interaction between the action control and the resource engine, let alone any alleged negotiation between the action control and the resource engine. Even further, the notices are sent in reaction to a change in a schedule’s condition, not as part of any alleged negotiation to form that schedule’s condition.

Section 7 (page 17) of the final Office Action further states the following:

“In addition, part of the negotiation processes also shown in Col. 29, lines 56-61 of Parad. Here, the negotiation occurs between the user and the intelligence of the action control via

logic and where user established rules in database through use of a menu.”

Col. 29, lines 56-61 of Parad states the following:

“A menu provides maintenance functions that construct reference tables of FIG. 11 to establish the rules for the behavior desired for each user. Much of the apparent intelligence of the action control is achieved by simple logic and data in these tables.”

This portion of Parad merely relates to the storage of something similar to a user profile used to personalize various application programs (like Microsoft Outlook™ for example). What is stored in the reference tables in Fig. 11 is a user-defined set of actions that might be performed in response to a given message type. They represent a list of user-specified actions. What the user chooses to store has nothing to do with conditions established by negotiation.

As noted in section 17 of the Office Action, page 11, lines 22-25 of the specification defines the term negotiation as follows: “The mechanism for making SLAs is negotiation - a joint decision making process in which the parties verbalise their (possibly contradictory) demands and then move towards agreement by a process of concession or search for new alternatives.” Parad does not disclose any such negotiation. Indeed, the unilateral entry of rules by a user does not involve any negotiation. The user’s unilateral entry of rules does not involve a joint decision making process in which the user and another party verbalize their possibly contradictory demands and then move towards agreement by a process of concession or search for new alternatives. There is no concession in the user’s entry of rules.

Col. 4, lines 52-56 of Parad states “The present invention contemplates that decisions are distributed throughout an enterprise and that resource conflicts must be identified, prioritized by importance, alternatives analyzed, and corrective actions implemented.” This general disclosure of Parad does not teach or suggest establishing conditions applicable to the provision of component processes in a service through negotiation. Appellant submits that this portion of Parad generally relates to determining an appropriate schedule in response to the requirements fed from external enterprise information systems 214 to resource engine 203.

Even assuming *arguendo* that negotiation is taught or suggested by Parad as alleged by the final Office Action, there is no further teaching or suggestion in Parad (and/or Wrabetz) of establishing conditions for provision of one or more component processes proactively by negotiation prior to receipt of a service request, as further required by independent claims 53 and 61. Section 7 of the final Office Action states “Here, the requirements are fed into the Resource Engines component where a schedule that meets requirements is attempted to be created. This attempt represents the negotiation process since the Resource Engine tries to meet specific requirements requested by the requester by matching different resources with each request.” As discussed above, the requirements fed into the resource engine are from the external enterprise information system 214. The requirements, if anything, therefore form a service request. That is, the requirements form a request to determine (and/or adjust) a schedule. Accordingly, if this “attempt” of feeding requirements into the resource engine to create a schedule “represents the negotiation process” itself as alleged by the Office Action, it is not possible for the negotiation to occur prior to receipt of the service



request (the receipt of the requirements). Accordingly, the interpretation expressed in section 7 of the final Office Action teaches away from a negotiation occurring prior to the receipt of the service request. This prior negotiation of rules by the invention speeds up the system's response to a service request. This benefit is not appreciated at all by Parad or Wrabetz.

Accordingly, the combination of Parad and Wrabetz thus fails to teach or suggest negotiation as claimed, let alone negotiation prior to receipt of a service request. Appellant thus requests that the rejection of claims 53-58 and 61-66 under 35 U.S.C. §103 be reversed.

Claims 68-76 are not made "obvious" under 35 U.S.C. §103 over Parad and Babayev.

Independent claims 68 and 76 require, *inter alia*, "the execution of said autonomous software processes establishing conditions applicable to provision of said service by negotiation between said first and second autonomous software processes." Independent claim 74 requires "a) establishing conditions applicable to provision of said service by negotiation between said autonomous software process and one or more other autonomous software processes." Section 6 (page 10) of the final Office Action alleges that col. 29, lines 49-61 of Parad teaches this claim feature. Appellant respectfully disagrees for reasons similar to that discussed above with respect to independent claims 53 and 61. Babayev fails to remedy this deficiency of Parad.

Moreover, page 10, lines 7-20 of the final Office Action alleges that the action control disclosed by Parad forms the claimed first autonomous software process. However, page 11, lines 16-17 of the Office Action admits that "Parad fails to disclose

executing a second autonomous software process....” Accordingly, the allegations of the Office Action that Parad discloses the claimed negotiation is clearly erroneous since the negotiation is between the first and second autonomous processes. That is, the final Office Action’s admission that Parad fails to disclose a second autonomous process contradicts that final Office Action’s allegation that Parad discloses negotiation between the first and second autonomous processes. Two autonomous processes are necessary to negotiate. The final Office Action admittedly only identifies one of the autonomous processes and therefore Parad cannot possibly teach negotiation under its own characterization of Parad.

The combination of Parad and Babayev further fails to teach or suggest “the execution of said autonomous software processes establishing conditions applicable to provision of said service by negotiation between said first and second autonomous software processes...the execution of the first autonomous software process subsequently involving the receipt of a request for said service for handling by said first autonomous software process,” as required by claim 68 and 76. Independent claim 74 requires a similar limitation. In Babayev, in response to a customer service request, the customer is immediately given a broad time window in which service provider personnel will turn up at the customer’s premise. Babayev thus generates a broad time window in reaction to a customer service request. There is no teaching or suggestion in Babayev (and/or Parad) of negotiation between autonomous software processes, and subsequent receipt of a request for service.

Appellant thus requests that the rejection of claims 68-76 under 35 U.S.C. §103 be reversed.

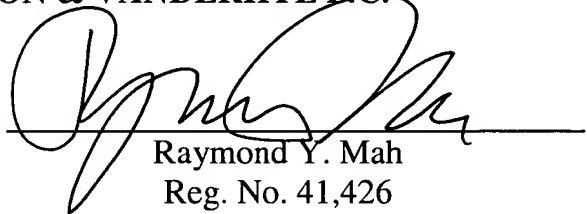
**CONCLUSION**

In conclusion it is believed that the application is in clear condition for allowance; therefore, early reversal of the Final Rejection and passage of the subject application to issue are earnestly solicited.

Respectfully submitted,

**NIXON & VANDERHYE P.C.**

By:



Raymond Y. Mah  
Reg. No. 41,426

RYM:sl  
901 North Glebe Road, 11th Floor  
Arlington, VA 22203-1808  
Telephone: (703) 816-4000  
Facsimile: (703) 816-4100

(VIII) CLAIMS APPENDIX

1-52. (canceled)

53. A service provision system for use in distributed processing environments, said system comprising:

an input for receiving a service request;

processing means for processing the service request;

negotiation means for use in establishing conditions applicable to provision of one or more component processes involved in provision of the service, said negotiation means being adapted to assemble said conditions proactively by negotiation prior to receipt of said service request;

an up-datable data store;

means to access said up-datable data store for storing said conditions when established and assembled; and

an output for providing a response to the service request, said response comprising an indication of availability of the requested service;

wherein the processing means is adapted to process a service request by accessing one or more of the previously established conditions in the data store, processing the request using the one or more established conditions and producing said response.

54. A service provision system as in claim 53 wherein one or more of said established conditions has an associated expiry time after which it is no longer applicable.

55. A service provision system as in claim 54 wherein the processing means is adapted to detect an expired or undefined condition in the data store, which condition is applicable to a component process for the provision of a requested service, and to trigger the negotiation means to establish a substitute condition.

56. A service provision system as in claim 55 further comprising:  
means to access said data store for storing data related to services offered by the system and to one or more entities which have an interest in receiving information relating to one or more of said services, together with means to transmit information based on said data related to services to the one or more entities which have an interest.

57. A system as in claim 53 which further comprises initiation means to initiate one or more component processes in provision of a requested service.

58. A service provision system as in claim 57 wherein:  
provisioning a requested service requires provision of a selected set of component processes;

the negotiation means establishes and stores a set of conditions applicable to provision of the component processes of the selected set; and

the processing means is adapted to process a service request by accessing the stored set of conditions in the data store, processing the request using said stored set, and producing said response.

59-60. (canceled)

61. A service provision method for use in distributed processing environments, said method comprising:

establishing conditions applicable to provision of one or more component processes in a service, proactively by negotiation prior to receipt of said service request;

accessing an up-datable data store and storing said conditions once established;

subsequently receiving a request for said service;

processing said service request by:

a) accessing one or more of said previously established conditions in the data store; and

b) providing a response to the service request, said response comprising an indication of availability of the requested service dependent upon whether said one or more established conditions is met.

62. A service provision method according to claim 61 wherein one or more of said established conditions stored in said data store is applicable until advent of an expiry time associated with said one or more conditions.

63. A service provision method according to claim 62 wherein further comprising the step, responsive to receipt of said request, of finding whether any conditions for provision of said service are extant and substituting a substitute condition in the event that no such conditions are found.

64. A service provision method according to claim 61 wherein said method further comprises the step of scheduling provision of said one or more component processes, said step being carried out after receipt of said request for said service.

65. A service provision method according to claim 64 wherein said method further comprises the step, responsive to a failure to schedule one or more component processes, of carrying out one of the following steps:

- i) re-schedule the component process;
- ii) transmit a message to an entity which requested the service indicating that the service can only be provided under conditions different to said previously established conditions;
- iii) re-assign the service to another service provider; or
- iv) indicate to an entity which requested the service that the requested service cannot be provided.

66. A method according to claim 61 further comprising the step of identifying component processes for use in provisioning the requested service.

67. A method according to claim 66 which further comprises initiating one or more of said component processes identified for use in the requested service.

68. A method of operating one or more computers to manage a business process, said method comprising:

executing a first autonomous software process;

executing a second autonomous software process;

the execution of said autonomous software processes establishing conditions applicable to provision of said service by negotiation between said first and second autonomous software processes;

accessing an up-datable data store and storing said conditions in said data store once established;

the execution of said first autonomous software process subsequently involving the receipt of a request for said service for handling by said first autonomous software process; and

responsive to receipt of said request, said first autonomous software process testing whether the established conditions for provision of said service are met and providing an indication as to whether the requested service is available in dependence upon the outcome of said test.

69. A method according to claim 68 in which one or more of said established conditions is stored in said data store until advent of an expiry time associated with said one or more conditions.

70. A method according to claim 68 wherein, responsive to receipt of said request, said first autonomous software process tests whether any conditions for provision of said service are extant and substitutes a substitute condition in the event that no such conditions are found.



71. A method according to claim 69 wherein, responsive to receipt of said request, said first autonomous software process tests whether any conditions for provision of said service are extant and substitutes a substitute condition in the event that no such conditions are found.

72. A method according to claim 68 in which said first autonomous software process identifies one or more further autonomous software processes representing resources required to provide one or more component services in the provision of said service, said method further comprising executing said further autonomous software process representing a component service provider.

73. A method according to claim 69 in which said first autonomous software process identifies one or more further autonomous software processes representing resources required to provide one or more component services in the provision of said service, said method further comprising executing said further autonomous software process representing a component service provider.

74. An apparatus for representing a provider of a service component of a business process, said apparatus comprising:

a computer having a memory storing autonomous software code executable to provide an autonomous software process representing a service requester, the execution of said autonomous software process:

- a) establishing conditions applicable to provision of said service by negotiation between said autonomous software process and one or more other autonomous software processes;
- b) accessing an up-datable data store and storing said conditions in said data store once established; and
- c) on subsequently receiving a request for said service, accessing said data store and testing whether the established conditions for provision of said service are met and providing an indication as to whether the requested service is available in dependence upon the outcome of said test.

75.<sup>1</sup> Apparatus according to claim 72 in which said autonomous software code is executable to provide a plurality of software threads, able to concurrently interact with different other autonomous software processes.

76. A method of operating one or more computers to manage a business process, said method comprising:

- executing a first autonomous software process;
- executing a second autonomous software process;
- the execution of said autonomous software processes establishing conditions applicable to provision of said service by negotiation between said first and second autonomous software processes;

---

<sup>1</sup> As noted above, claim 75 should correctly depend from claim 74, not claim 72.

accessing an up-datable data store and storing said conditions in said data store once established;

the execution of said first autonomous software process subsequently involving the receipt of a request for said service for handling by said first autonomous software process; and

responsive to receipt of said request, said first autonomous software process testing whether the established conditions for provision of said service are met and providing an indication as to whether the requested service is available in dependence upon the outcome of said test;

wherein there are no control dependencies between the first and second autonomous software processes.

O'BRIEN et al  
Serial No. 09/043,406

**(IX) EVIDENCE APPENDIX**

None

O'BRIEN et al  
Serial No. 09/043,406

(X) RELATED PROCEEDINGS APPENDIX

None